ADAPTIVE DEBLOCKING AND DERINGING OF H.264/AVC VIDEO SEQUENCES

Ehsan Nadernejad, Nino Burini, Søren Forchhammer

Department of Photonics Engineering, Technical University of Denmark, Ørsteds Plads,
2800 Kgs. Lyngby, Denmark

ABSTRACT

We present a method to reduce blocking and ringing artifacts in H.264/AVC video sequences. For deblocking, the proposed method uses a quality measure of a block based coded image to find filtering modes. Based on filtering modes, the images are segmented to three classes and a specific deblocking filter is applied to each class. Deringing is obtained by an adaptive bilateral filter; spatial and intensity spread parameters are selected adaptively using texture and edge mapping. The analysis of objective and subjective experimental results shows that the proposed algorithm is effective in deblocking and deringing low bit-rate H.264 video sequences.

***Index Terms*—** deblocking, deringing, H.264, Bilateral Filter, Post-processing

1. INTRODUCTION

Block based video codecs like MPEG-4 [1] and H.264/AVC [2] may suffer from ringing and blocking artifacts, which require effective post-processing to be reduced. Post-processing improves image quality without changing existing standards.

Many deblocking and deringing algorithms have been proposed for compressed images and videos [3–13]. Kim [3] proposed an adaptive deblocking algorithm for low bitrate video, where the DC and AC values are used to label each block as low or high activity; then, based on the classification, two kinds of low-pass filters are adaptively applied on each block. A method combining the directional anisotropic diffusion equations with adaptive fuzzy filtering for removing blocking and ringing artifacts was presented in [4]. Zhai proposed an algorithm for deblocking [9], consisting of three parts: local AC coefficient regularization (ACR) of shifted blocks in the discrete cosine transform (DCT) domain, blockwise shape adaptive filtering (BSAF) in the spatial domain, and a quantization constraint (QC) in the DCT domain. Yao et al. [8] introduced an algorithm using histogram driven diffusion coefficients for post-processing.

This work introduces a new algorithm to reduce blocking and ringing artifacts in H.264 video sequences. Deblocking is done with a decision mode-based algorithm using local characteristics of the blocks and a quality metric of each frame (I, B, P). After deblocking, an adaptive bilateral filter is applied to the regions with ringing artifacts. The experimental results show that the proposed algorithm effectively reduces blocking and ringing, outperforming other methods with respect to PSNR, MSSIM and subjective tests.

The rest of this paper is organized as follows. The proposed algorithm is described in Section 2. Section 3 shows the experimental results on H.264/AVC video sequences. Finally, we conclude in Section 4.

2. PROPOSED ALGORITHM

The proposed algorithm consists of two steps: deblocking and deringing. In the first step, the quality of each frame (I, P, B) is calculated using a quality metric and deblocking is done using decision modes. In the second step, a bilateral filter with adaptive spatial and intensity spread parameters is applied to the deblocked image for deringing.

The deblocking scheme is based on region classification with respect to the activity across block boundaries; depending on the classification, three different filtering modes are applied in the horizontal and vertical directions. Hard filtering is used on flat areas, whereas weak filtering is used to preserve sharpness in areas of high spatial or temporal activity. An intermediate mode (without filtering) is used to solve the problem of too coarse a decision and avoid either excessive blurring or inadequate removal of the blocking effect. Figure 1 presents a flowchart of the proposed algorithm.

2.1. Deblocking

In the deblocking step, the decision mode is done based on a pixel quality metric and predefined thresholds. Appropriate two steps filtering is then performed based on decision modes.

2.1.1. Quality Measure for Pixels in H.264 sequence

A compressed video sequence is mainly degraded by coarse quantization of DCT coefficients and inaccurate motion compensation. Due to different quantization steps and different frame types (I, P, B), pixels are distorted with different degrees and providing different qualities. Based on quantization step size and frame types, we estimate a quality parameter for each pixel which is used in the decision mode step.

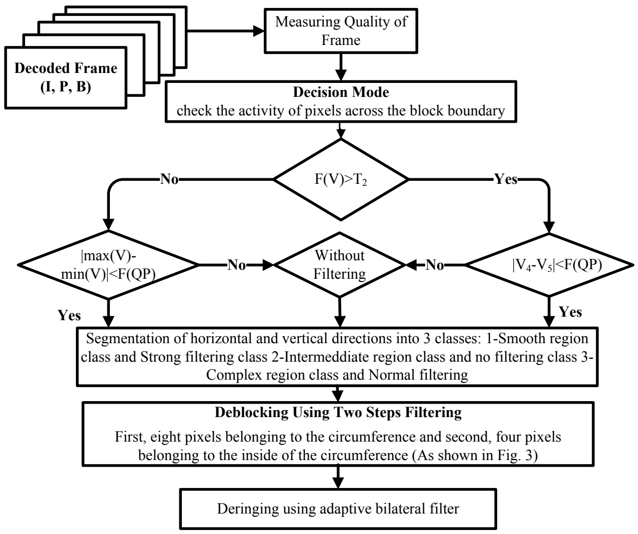


Fig. 1: Flowchart of the proposed algorithm.

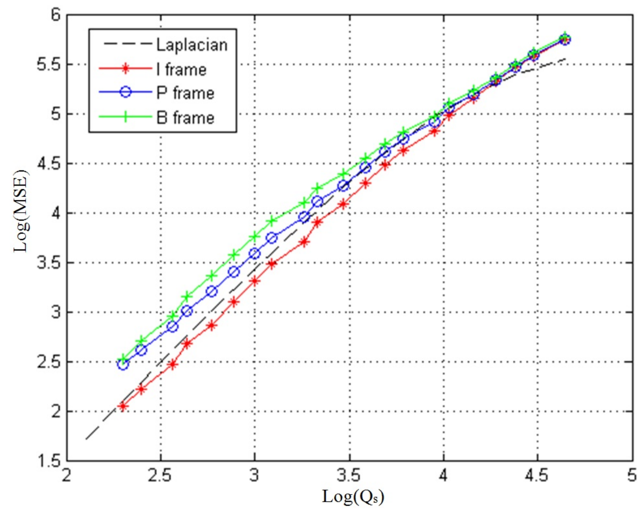


Fig. 2: MSE vs Q_s measured on mobcal (CIF); rate control is disabled, different QP values chosen for the different points [11].

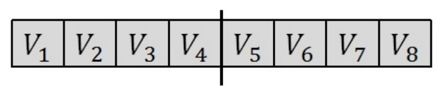


Fig. 3: Position of filtered pixels and pixel vector used for the decision mode process in horizontal direction. For vertical direction the pixel vector is the same.

The quality measure (Q_M) is defined to reflect approximate MSE for each pixel in I, P and B frames [10]:

$$Q_M = \sqrt{12 \times MSE} \tag{1}$$

Pixels with smaller Q_M values are considered to have higher quality. This pixel quality parameter cannot reflect the quality of each individual pixel accurately, and it is just used to compare approximately individual pixels with different quantization step and frame type [10, 11]

The curves shown in Fig.2 were obtained by measuring the MSE of the luminance components of H.264/AVC decoded sequences. QP determines the quantizer step size [11]. The Laplacian distribution is used to model the MSE quality as shown in Fig.2. The results indicate that intra coded frames (I) provide the best quality, and that unidirectional prediction frame (P) have better quality than bidirectional prediction frame (B). As QP increases, degradations for I, P and B frames are all increasing, while the quality differences among I, P and B frames are decreasing. In this paper, these training data are used to describe relative comparisons between different coding modes. All the settings and testing in later experiments are based on these curves. With the QP value and frame type we can calculate the quantization step size (Q_s) and use Fig.2 to get an MSE estimate which provided Q_M using Eq.1. The decision modes and segmentation step use the following function of Q_M:

$$F(QP) = \sqrt{Q_M} \tag{2}$$

2.1.2. Decision Modes and Segmentation

This step classifies the pixels activity in the regions to be filtered and applies the appropriate filter depending on the features of the region. The filtering modes are determined based on the variation of activity in vertical and horizontal pixel vectors at each 4×4 block boundary, as shown in Fig. 3.

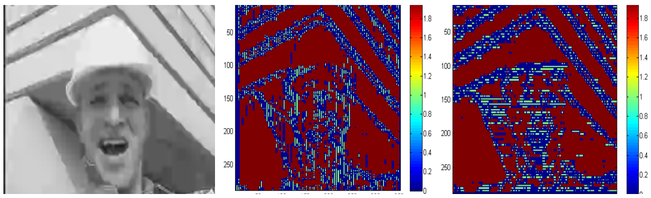
In this step, flat regions and complex regions are classified by local characteristics. An activity factor is assigned to the pixels inside each vector of pixels at the 4×4 block boundaries, as described in Fig. 3. The activity is as follows:

$$R(V) = \sum_{i=1}^{7} \phi(v_i - v_{i+1}), \tag{3}$$

where $0 \leq R(V) \leq 7$ and

$$\phi(\Delta) = \begin{cases} 1, & \text{if } |\Delta| < T_1 \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where T_1 is a fixed threshold (should be set to a small value), V represents the eight-pixels vector and v_i are the pixel values. The activity factor $R(V)$ reflects the activity in V across block boundary; it also represents the number of detected edges inside V. If the value of $R(V)$ is smaller than a certain threshold T_2, and the difference between the maximum and minimum values of V is smaller than $F(QP)$, we assume V to be in a complex region and apply the filter for complex region. If $R(V)$ is bigger than $F(QP)$, then it is does not need filtering. If $R(V) > T_2$, the two pixel values on either side of the block boundary (v_4 and v_5) are considered. If the absolute



a b c

Fig. 4: Decision mode of the 25^{th} frame of the Foreman sequences, a) coded frame, b) horizontal direction modes, c) vertical direction modes.

a_1	a_2	a_3	a_4	a_5	a_6
a_7	y_1	x_1	x_2	y_2	a_8
a_9	x_5	y_3	y_4	x_7	a_{10}
a_{11}	x_6	y_5	y_6	x_8	a_{12}
a_{13}	y_7	x_3	x_4	y_8	a_{14}
a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}

Fig. 5: Pixels must be filtered in two steps filtering in window (6×6).

difference of two pixels is smaller than $F(QP)$, we assume V to be in a smooth region, otherwise it does not need filtering. In this work, $T_1 = 6$ and $T_2 = 2$.

Based on the decisions mode in horizontal and vertical directions, the frame is segmented in three no filtering (N), weak filtering (W) and hard filtering (H) regions. Figure 4 shows an example of segmentation.

2.1.3. *Two steps filtering for deblocking*

Two steps filtering is done after segmentation and labeling of each pixel. A 6×6 filtering window is centered at the intersection of four 4×4 pixel blocks as shown in Fig. 5. The filtering window is placed at the upper left corner of the frame and is shifted across the whole frame.

Deblocking is done in two steps. In the first step, only eight pixels are filtered at the intersection of four 4×4 pixel blocks $(x_1 \ldots x_8)$. As mentioned before, there are two options for each pixel in the both vertical and horizontal directions. After segmentation, if no filtering mode is selected in any direction with other filtering modes, only one dimensional filters are required. For instance, in NW or WN modes just apply a weak 1D filter on the target pixels in vertical and horizontal directions, respectively. If NH or HN is selected then a hard 1D filter is applied to the target pixel in one of both directions. When the filtering mode belongs to the weak filtering and hard filtering mode (WH, HW, HH), 2D filtering is applied on the pixel. Equation 5 shows the updated values of the x_{1u} in different modes ($\gg$ is the bitshift operator).

$$x_{1u} = \begin{cases} y_1 + 5x_1 + 3x_2 - y_2 \gg 3 & \text{if NW,} \\ 2a_7 + y_1 + 2x_1 + x_2 + y_2 + a_8 \gg 3 & \text{if NH,} \\ 4(x_1 + a_3) + 2(x_2 + y_2 + y_3) + y_1 + y_2 \gg 4 & \text{if WH,} \\ 4(a_2 + a_3 + a_4 + x_1) + 2(y_1 + x_5 + y_4 & \\ \quad +y_6) + a_7 + x_2 + y_2 + a_8 + y_3 + y_5 + & \text{if HH.} \\ \qquad\qquad x_3 + x_6 \gg 5 & \end{cases} \tag{5}$$

The other pixels are filtered in the same way. To limit computation, the weighting matrix of the 2D filter is simplified and some coefficients are cut or rounded. The literature includes different methods for simplification [3, 5].

At the end of the first step, pixels belonging to $x_1 \ldots x_8$ are filtered and in the second step, the remaining pixels belonging to $y_3 \ldots y_6$ are filtered by applying the appropriate filter according to the pre-assigned filtering mode. The pixels will update according to their filtering mode as follows:

$$y_{3u} = \begin{cases} 3x_5 + y_3 + y_7 - y_4 \gg 2 & \text{if NH} \\ 2x_5 + 5y_3 + 3y_4 - 2x_8 \gg 3 & \text{if NW} \\ 6x_5 + 4y_3 + 2y_4 + 4x_7 + 2y_5 - x_1 - x_3 \gg 4 & \text{if HW} \\ x_5 - x_7 + x_1 - x_3 + 4y_3 + 2y_5 + 2y_4 \gg 3 & \text{if WW} \\ 2(x_5 + x_7 + x_1 + x_3) + y_5 + 6y_3 + y_4 \gg 4 & \text{if HH} \end{cases} \tag{6}$$

For symmetric filtering modes, the filtering values are simply computed in a symmetric manner.

2.2. Bilateral Filter for Deringing

After removing the blocking artifacts from the frame, an adaptive bilateral filter is used to remove ringing artifacts. The bilateral filter is a nonlinear weighted averaging filter, obtained by combining two Gaussian filters; one filter works in spatial domain, other filter works in intensity domain [14]. The weights depend on both the spatial distance and the intensity distance with respect to the center pixel. The main feature of the bilateral filter is its ability to preserve edges while doing spatial smoothing. At pixel location x, the output of a bilateral filter can be formulated as follows:

$$J(x) = \frac{1}{Z} \sum_{y \in \psi(x)} e^{\frac{-\|(y-x)\|^2}{2\sigma_d^2}} e^{\frac{-|(I(y)-I(x))|^2}{2\sigma_r^2}}, \tag{7}$$

where σ_d and σ_r are parameters controlling the fall-off of weight in spatial and intensity domains, respectively. $\psi(x)$ is the spatial neighborhood of pixel $I(x)$ and Z is a normalization constant:

$$Z = \sum_{y \in \psi(x)} e^{\frac{-\|(y-x)\|^2}{2\sigma_d^2}} e^{\frac{-|(I(y)-I(x))|^2}{2\sigma_r^2}}. \tag{8}$$

The behavior of the bilateral filter is determined by σ_d and σ_r. For deringing, these parameters should be chosen carefully, since it is desirable to avoid over-smoothing texture regions and to preserve edges in edge regions. These could be done first by estimating the texture regions and discontinuity of the edges, and then control the extent of smoothing and sharpening through the σ_d and σ_r values. In the proposed method, each 4×4 block is classified into one of the four categories: strong edge, weak edge, texture and smooth blocks. For a smooth region, the value of the σ_d can be large, otherwise it should be small. Classification is done by computing the standard deviation (STD) in a 4×4 window around each pixel and comparing the maximum STD in each 4×4 block with a set of predetermined thresholds as follows:

$$\sigma_d = \begin{cases} StrongEdge, \sigma_d = 0.8 & \text{if MaxSTD} \in [35, \infty) \\ WeakEdge, \sigma_d = 1.8 & \text{if MaxSTD} \in [25, 35) \\ Texture, \sigma_d = 2.8 & \text{if MaxSTD} \in [15, 25) \\ Smooth, \sigma_d = 3.8 & \text{if MaxSTD} \in [0, 15) \end{cases} \quad (9)$$

The optimal σ_r value of the bilateral filter is linearly proportional to the standard deviation of the noise ($\sigma_r = \alpha \times \sigma_n$). The noise variance is estimated with the robust median noise estimator technique [11]. In the proposed algorithm, the value of α is set to $1/3$ in each 4×4 block. The calculation of σ_d and σ_r are repeated for all blocks to obtain the block spatial map M_{σ_d} and the block intensity map M_{σ_r}.

3. SIMULATION RESULTS

The performance of the proposed algorithm was evaluated on H.264/AVC video sequences through comparison with our implementation of several state-of-the-art spatial post-processing algorithms [3–9, 12, 13][1]. The GOP structure was defined as $(IPPB)_{12}$. Two different types of experiments have been performed. In the first experiment, the algorithm was applied with two different quantization parameters ($QP = 35, 45$) with the in-loop deblocking filter enabled. In the second experiment the in-loop deblocking filter was disabled. Several CIF (4:2:0) test sequences were chosen: Akiyo, Bus, Coastgard, Container, Cycling, Foreman, Hall, Mobcal, Mother and Daughter. The algorithms were applied on the first 100 frames of each sequence. The qualities of the different algorithms have been compared in terms of Weighted-PSNR and Weighted-MSSIM, where the luma and chroma components have a weight of 2/3 and 1/6, respectively [15, 16]. The comparative objective results for the first experiment are summarized in Table 1. It can be seen that the proposed algorithms achieves higher PSNR and MSSIM compared to the other algorithms.

In the second experiment, the H.264/AVC in-loop deblocking filter was disabled. The proposed algorithm reaches higher PSNR and MSSIM when compared to the in-loop filter alone. Table 2 shows the performance of the proposed algorithm against H.264/AVC when the in-loop filtering is disabled on Akiyo video sequences.

Figure 6 visually compares the in-loop filter and the proposed post-processing algorithm for deblocking and deringing. It can be seen that the blocking and ringing artifacts are more effectively attenuated in both images, resulting in a better perceptual quality for the reconstructed video.

[1]The software for [9] was provided by Zhai

Table 1: The average results of post-processing H.264/AVC video test sequences using different algorithms.

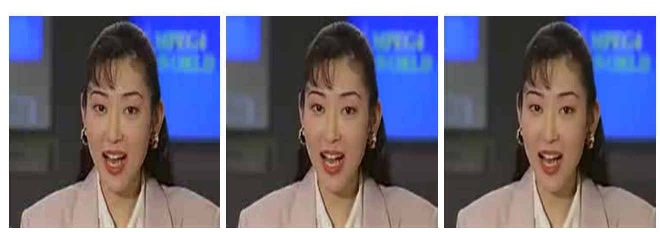
Metric	PSNR		MSSIM	
QP	35	45	35	45
H.264/AVC	34.76	30.51	0.906	0.810
[3]	34.09	30.35	0.898	0.806
[4]	35.09	30.63	0.911	0.812
[5]	35.03	30.52	0.910	0.811
[6]	34.66	30.49	0.901	0.809
[7]	34.82	30.57	0.907	0.813
[9]	35.03	30.59	0.910	0.813
[8]	35.00	30.36	0.907	0.809
[12]	35.04	30.39	0.909	0.810
[13]	34.82	30.44	0.908	0.809
Proposed	35.18	30.62	0.911	0.814

Table 2: Results of H.264/AVC video when the in-loop filtering is disabled/enabled and with the proposed algorithm on Akiyo video.

Metric	PSNR		MSSIM	
QP	35	45	35	45
Disabled in-loop	33.15	28.21	0.895	0.793
Enabled in-loop	33.25	28.36	0.912	0.806
Proposed	33.30	28.43	0.912	0.812

4. CONCLUSION

We have proposed an adaptive post-processing algorithm for blocking and ringing artifact reduction in H.264/AVC video sequences. The algorithm uses a quantization parameter to estimate the quality of each frame. Deblocking is performed using a quality metric and the activity of pixels across of the block boundary; a deringing algorithm is applied to the areas which have ringing artifacts using an adaptive bilateral filter. Results show that the proposed algorithm improves the objective and subjective quality of H.264 video sequences.



a b c

Fig. 6: The comparison of filter result on Akiyo (75^{th} frame), a) Compressed frame, b) In-loop filter, c) Proposed algorithm.

5. REFERENCES

[1] "MPEG-4 Video Verification Model Version 8.0," MPEG Video Group, ISO/IEC JTC1/SC29/WG11 N1796, July 1997.

[2] "Draft ITU-T Recommendation and Final Draft International Standard Joint Video Specification," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, ITU-T Rec. H.264–ISO/IEC, 14496-10 AVC JVT G050, 2003.

[3] Changick Kim, "Adaptive post-filtering for reducing blocking and ringing artifacts in low bit-rate video coding," *Sig. Proc.: Image Comm.*, vol. 17, no. 7, pp. 525–535, 2002.

[4] Ehsan Nadernejad, Sren Forchhammer, and Jari Korhonen, "Artifact reduction of compressed images and video combining adaptive fuzzy filtering and directional anisotropic diffusion," in *3rd European Workshop on Visual Information Processing (EUVIP), Paris, July 4-6*, 2011, pp. 24–29.

[5] Salim Chebbo, Philippe Durieux, and Batrice Pesquet-Popescu, "Adaptive Deblocking Filter for DCT Coded Video," in *Proc. Fourth Int. Workshop on Video Process. and Quality Metrics for Consumer Electr.*, Jan 2009.

[6] Shen-Chuan Tai, Yen-Yu Chen, and Shin-Feng Sheu, "Deblocking filter for low bit rate MPEG-4 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 733 – 741, June 2005.

[7] Hao-Song Kong, Yao Nie, A. Vetro, Huifang Sun, and K.E. Barner, "Adaptive fuzzy post-filtering for highly compressed video," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, oct. 2004, vol. 3, pp. 1803 – 1806 Vol. 3.

[8] Susu Yao, Keng Pang Lim, Xiao Lin, and S. Rahardja, "A post-processing algorithm using histogram-driven anisotropic diffusion," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, may 2005, pp. 4233 – 4236 Vol. 5.

[9] Guangtao Zhai, Wenjun Zhang, Xiaokang Yang, Weisi Lin, and Yi Xu, "Efficient Deblocking With Coefficient Regularization, Shape-Adaptive Filtering, and Quantization Constraint," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 735 –745, Aug 2008.

[10] Bo Martins and Søren Forchhammer, "A unified approach to restoration, deinterlacing and resolution enhancement in decoding MPEG-2 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 803–811, 2002.

[11] Xing Huang, Huiying Li, and Søren Forchhammer, "A Multi-Frame Post-Processing Approach to Improved Decoding of H.264/AVC Video," in *Proc. IEEE ICIP*, 2007, vol. 4, pp. 381–384.

[12] Ming Zhang and Bahadir K. Gunturk, "Compression artifact reduction with adaptive bilateral filtering," in *Proc. SPIE*, 2009, vol. 7257, pp. 72571A–72571A–11.

[13] Sung Deuk Kim, Jaeyoun Yi, Hyun Mun Kim, and Jong Beom Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 156 –160, Feb 1999.

[14] Ming Zhang and Bahadir K. Gunturk, "Multiresolution Bilateral Filtering for Image Denoising," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2324 –2333, Dec 2008.

[15] Francesca De Simone, Daniele Ticca, Frederic Dufaux, Michael Ansorge, and Touradj Ebrahimi, "A comparative study of color image compression standards using perceptually driven quality metrics," in *Proc. SPIE Optics and Photonics, San Diego, CA USA, August 11-14*, 2008.

[16] Ehsan Nadernejad, Jari Korhonen, Søren Forchhammer, and Nino Burini, "Enhancing perceived quality of compressed images and video with anisotropic diffusion and fuzzy filtering," *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 222–240, 2013.